THE ELECTRIC SECRETARY
by
The Peripheral People

OPERATING INSTRUCTIONS

INTRODUCTION

The ELECTRIC SECRETARY program was written specifically to adapt TRS-80 hobby and business system to word processing applications. It requires a single or multiple disk system with a minimum memory of 32K. However, without modification, this configuration will print upper case letters only.

While this is satisfactory for many commercial applications, the TRS-80 can be easily modified for upper/lower case printing and video display. With the addition of a suitable daisy wheel printer (Qume, Diablo, Multi-Term, etc.), the "lowly" TRS-80 will outperform an IBM "Mag-Card II" $10,000 word processing system, in addition to its primary function as a computer.

ADVANTAGES

Writing the program in basic offers numerous advantages over machine language word processing programs. The operator can more clearly understand how the program works, by listing it. The operator also has a degree of control that is not available with machine language programs. For example, default values for margins, page length and so on can be modified by the operator for most often used page formats.

The ELECTRIC SECRETARY incorporates a hyphenating dictionary that is not found in machine language programs. In typical text, sooner or later, a word will come along that must be hyphenated. Lesser programs simply leave a gap at the end of the line when this occurs or else leave big gaps between words to fill out the line. This destroys the symmetry of a beautifully formatted page of text.

Another advantage of the ELECTRIC SECRETARY is the use of disk storage rather than cassettes. Once the text is completely entered, it is saved on disk as a basic line numbered program. This file can then be loaded from the disk into memory and totally manipulated using the TRS-80 edit commands. Words or sentences can be added, deleted, typos corrected and on. Once the operator is satisfied with the text, it can be resaved on disk for later use.

If the text is entered correctly, the operator can go directly to the print program without editing. Editing probably would not be required with short letters and correspondence can be quickly generated by the proficient operator.

One of the most useful features of the ELECTRIC SECRETARY is the ability to generate form correspondence. This is facilitated by a technique of coupling and cross coupling files. The user generates a file containing names and addresses and a second file consisting of the form letter. The files are cross coupled through commands and individual inside addresses are added to each form letter until the end of the address file is reached. If continuous letterhead rolls are used, no operator intervention is required during printing. Other files can be made from standard paragraphs and used to customize letters or to insert routine text in legal or other documents.

One unique feature not found in other TRS-80 word processing programs is an echo routine which permits the printer to be used as an electric typewriter. If desired, keyboard entries can appear on the screen and printout at the same time. This routine can also be used with other TRS-80 programs.

Another exclusive feature is an upper case shift lock to permit quantities of upper case text to be entered without continually depressing the shift key. This feature is only applicable when the upper/lower case modifications have been made.

BACKUP DISKS

Since you could make a serious error during some of the following steps (that might damage the program), make a backup copy before proceeding. Use this backup copy for your experimenting.

UP AND RUNNING

The ELECTRIC SECRETARY works whether you have made the upper/lower case modification or not. In either instance, the power-up procedure is as follows:

enter 6 for the "HOW MANY FILES" request. You could have as many as six files open at any one time. If this is not accomodated, the program will "crash". Next, protect the memory by entering 48991 for a 32K system. Run the MENU program and enter selection 4. If you press the reset button for any reason, this procedure should be followed. If you do not have the upper/lower case modification installed, you can avoid running the case program by removing the poke statements near the beginning of the PREP and PRINT programs. However, this will prevent using the echo routine. Also, if you have an upper/lower case printer, the copy will be reversed (upper for lower and vice versa) if the case program is not run.

WRITING LETTERS

If you have no word processing experience, you will want to see the ELECTRIC SECRETARY do something right away. Okeh, we'll humor you. Load the PREP program by selecting #1 on the menu. When the computer asks for a file name, use your first name (8 letters maximum) so you can remember it. Answer the upper/lower question and the disk will operate. As soon as the screen clears, enter the sample letter you'll find near the end of this manual exactly as shown. Use the enter key at the end of each line. The layout on the screen should look exactly like the sample letter. The disk will run occassionally as you type the copy. This is normal.

As soon as you type .end, the disk will run for a final time and the ready prompt will return. Then type new and run the PRINT program. Once again, answer the questions (remember the file name?). The printer should spring to life and type the letter if you didn't make any mistakes.

Now that you've had your fun with the ELECTRIC SECRETARY, let's put 'er to work. We've prepared a demonstration program to show you why you typed those rather mysterious letters. They are called commands.

LEARNING THE COMMANDS

Note that during the following instruction steps, you will be told to run the PRINT program a number of times. It is not necessary to run the MENU each time. You can simply type RUN"PRINT" and accomplish the same thing.

If you have not made the case modifications (and do not remove the poke statements), answer NO or N when requested. If you type YES (or any word starting with a Y), you will see strange things on your display.

To see what the ELECTRIC SECRETARY can do, let's demonstrate the various commands. In doing so, you will learn how to apply these commands during your word processing work.

The ELECTRIC SECRETARY is such a powerful program that it is difficult to describe every feature in detail. The following instructions emphasize experimenting to increase proficiency in using the program.

Insert the disk and LOAD (not RUN) file DEMO. Type list, then using "shift @" to stop the program lines, note that each line number is followed by an apostrophe. This is TRS-80 shorthand for a remark which does not get printed. Every line must have this apostrophe. A program full of remarks will not RUN. This is why you use LOAD"DEMO" instead of RUN. The ELECTRIC SECRETARY, of course, strips off these apostrophes before printing.

Note that the first few lines are alpha numeric information preceded by a period. These are commands which are acted upon, but not printed. If the first character following the apostrophe is not a period, the line will be printed.

A few commands you should see are; .CM .LM5 .RM60 .PL60 .SP1 .BL1 .NF and .FL. A list of commands, and the function of each, is given near the end of this manual. Most commands have two letters preceded by a period. These commands are followed either by numbers (n) or data (d) unless they are switch commands such as .NF (no fill) or .FL (fill). The printout of the DEMO program illustrates how these commands are embedded in the text to control the printing of that file. It is important to note that although more than one command can occur on a line, commands cannot occur in the same line as text.

The first line of any text must be "0'". This tells the computer that it is an ASCII file. Any time you resave a file (for example, after appending or renumbering), make sure the first line is "0'" and that you SAVE "file name",A. The "A" saves the file in ASCII. Since we'll be doing quite a bit of editing of the DEMO file, always remember to type SAVE"DEMO",A.

The command .cm at line 10 is not really a command at all. It is the ELECTRIC SECRETARY version of REM. Anything on the line after .cm (comment) will not be printed or acted upon. Lines 20 and 30 specify the margins, while 40 specifies the page length. Line 50 tells the computer to justify the copy. Line

60 produces single spacing and line 70 is the NO FILL command. The command .IT5 imbedded in the text (lines 180, 340, 410 and 470), is a 5 space paragraph indent.

Now let's see what these commands and text do. Type RUN"PRINT", answer "DEMO" for the file request and "P" for the printout request. The disk will start up and the printer will activate and print the demo text. Note: If you have not made the case conversion (and have an upper/lower case printer), the cases will print reversed from normal.

LINE FILLING AND JUSTIFICATION .JU1 .FL .NF

The printing certainly does not resemble a typed manuscript, however. The lines are printed just as they were entered originally. To print usable copy, line filling is required. Line filling is the most basic feature of the ELECTRIC SECRETARY. This is the process of adding words to a line until one more word will exceed the specified right margin. This word and the following words are saved for printing on the next line. This allows the printed version of the text to have lines differing in length from those entered.

One of the major features of a word processing program is to even the right margin like the left margin. This prints the text in even columns and is called justification. The .jul justify command cannot be acted upon unless .fl (fill) is also invoked. The ELECTRIC SECRETARY justifies the right margins by adding blanks and determines how many blanks are needed between words to expand the line to the specified length. The program adds blanks as required from left to right and right to left on alternate lines to make the added blanks less noticeable.

Once again, load the "DEMO" file and change line 190 to read 190',FL to command the line filling function. Save the file (SAVE"DEMO",A), then print the program again. The printed copy should be perfectly justified on the right margin.

Incidentally, you can also review the text on the screen rather than the printer. Run the program again, but this time specify c (for crt) rather than p (printer). The copy will now scroll on the screen.

HYPHENATION .SY1

At times, a long word at the end of the line exceeds the specified line length, yet if it is moved to the next line, excessive blanks will be required to justify the right margin. In this case, it is necessary to hyphenate the word.

Note the first line of the third paragraph. Excessive blanks have been used to fill the line because the word 'processing' will not fit on the first line.

Let's turn on the hyphenation feature. Reload the "DEMO" file and add 10',SY1. This will tell the computer to hyphenate words as required. Resave the program (don't forget the ,A) and run the PRINT program again. .SY1

When the third paragraph is reached, the ELECTRIC SECRETARY will come across a word (processing) that needs to be hyphenated. The program will first look in a file called "DICTNARY". If the word is found, it will be divided at the appropriate point and printed. Since we've cleared the "DICTNARY" file, the word will not be found and the cassette control relay will buzz to alert you of this fact.

When the computer tells you that the word in not in its dictionary, type 'pro-cess-ing' and press enter. The program will proceed after this word has been stored in the DICTNARY file. The word manipulated will also have to be hyphenated later on in the text. Look this one up in the dictionary for the correct hyphenation points.

FORMATTING

The ELECTRIC SECRETARY program also provides for total control of page formatting. The most important function is margin control. The format commands are usually specified during the first few lines of the program. However, you can change margins and alter the page length within the body of the text. This can be used for special effects in your documents. Line spacing, justification and line fill can be turned on and off at any time while preparing the document.

Margin Control - The following are the margin control commands:

Page Length	(.PL(n)	66
Top Margin	(.TM(n)	0
Bottom Margin	(.BM(n)	6

Left Margin	(.LM(n)	0
Right Margin	(.RM(n)	70
Margin Adjust	(.MA(n)	MA.

A complete list of commands, and an explanation of their function, is given in a table near the end of this manual.

DEFAULT VALUES

Note that the original DEMO text did not specify top and bottom margins. The default values were used instead. To illustrate how this works, reload the DEMO program and delete lines 20, 30 and 40 to remove these format commands. Resave the program (,A) and print it. When no format commands are given, default values are used instead. The printout should now have a line width of 70 characters since the default left margin is zero and the right margin is 70. For most work there is no point of having a left margin other than zero. The paper can be physically positioned in the printer to provide a proper left margin. This will print text faster since the printer will start the line immediately without having to tab over (n) spaces.

The default values can be overridden at any time by imbedding suitable commands in the text. Reload the DEMO file. Insert line number 345 just before the third paragraph and type '.LM10.RM60. At line 405 insert the original default values: 405'.LM0.RM70. When you print this revised text, the third paragraph should be narrower and subsequent paragraphs should return to the default format.

Figure 1 shows the variable names assigned to various functions within the program. The default values for the page format are given in line 370 of the PRINT program. These values are set up for 10 point type. However, if a 12 point wheel or ball is used, the values should be changed accordingly. For example, P=66 would be changed to approximately 88 (8 lines/inch) to fill an 11 inch printed page. L2 (the right margin) should also be changed. By consulting the variable list (Figure 1) and line 370, virtually any default format can be selected.

SPACING .SP(n)

Reload the DEMO text, change line 50 to ',SP2. Resave and reprint the text. It should print double spaced. The spacing command can be imbedded in the text to switch back and forth between single and multiple line spacing, if desired.

SWITCH COMMANDS .FL .NF .WT1

Fill (.FL) and no fill (.NF) are switch commands. Note that they are used before and after text that you do not want to be filled. If this were not done, these words would be continued as if they were part of a continuous text.

There is another important switch command in the ELECTRIC SECRETARY program. Let's say you are printing on stationery. The operator would have to be rather agile to get a new page in the printer before printing commenced on the next page. However, there is a command to cover this. Reload the DEMO text, insert '.WT1 somewhere in the command area of the program. Resave and reprint. This time, the printer will stop and the screen will instruct you to insert and align a new sheet of paper. Press ENTER to proceed. You can also imbed the command .wa whenever you want the printing action to stop until you press the enter key.

DATA COMMANDS .CE .HD(d) .WA

So far, we have used number commands and switch (on-off) commands. The ELECTRIC SECRETARY also uses data commands. One of the most common is .CE to center a line on the printed page.

Reload the DEMO test and edit lines 80, 90, 100 and 130 to insert .CE between the apostrophe and the following letters (for example, 80'.CETHE ELECTRIC SECRETARY). This will tell the computer to center these lines where they should be to make the page look correct. Resave and reprint the edited program and note the centered titles.

Another important command that expects data is HEADER (.HD(d). Whatever is printed following the HEADER command will print at the upper left corner of the page. For example, if you insert 75'.HDMr. Jones-, each page following the first will print Mr. Jones- Page 2, and so on. If a HEADER is not given, the default is to print only the page number. If page numbers are not desired, the command .HDno will turn off any header including the page numbering routine. If the operator wants the printing to stop so that copy can be manually typed on the typewriter, imbed the command (.WA) on a line by itself. Then on the next line, continue the text. The printer will stop until the operator presses the enter key on the TRS-80. If you have a printer, rather than a typewriter, you will

have to use the TRS-80 keyboard to enter text. Make sure you energize the echo routine before printing text that will require insertions. For a better method of accomplsihing this, see "COUPLING FILES".

WRITING TEXT

Once you are satisfied you have a basic understanding of the commands, try processing your own text. Load the MENU program and run the upper/lower case program if you have not removed the poke statements as discussed earlier. Then select #1, the PREP program. The program will first ask you for a file name. For new text, select a suitable 4-5 letter name (do not use more than 8 letters!). If you specify an existing file name (for example, DEMO), the buzzer will alert you to the fact that this file name already exists. The program will then ask you if you want to overwrite or append (add to) it. If you select "O", the old file will be erased and a new file opened using this name. If you select "A", the existing text will be moved to a new file called TEMP and you can add to it. The program will start you at the next available line number from where you left off. When edited, you will have to add the "0'" and save the TEMP file under the original name selected for the file. The TEMP file is not renamed after you type .END. More about appending text later.

After the screen clears, type .SP1.SY1.JU1.FL and then enter some text. Make it a practice to press ENTER at the end of each sentence. Under no circumstances should you type more than 4 lines without pressing enter. If you do, you will overload the text buffer. This will "crash" the program and possibly even put "garbage" on some of the other disk tracks. Note that the disk runs every few lines as the buffer contents are moved to the disk for storage. This is the reciprocal of the action you noted when the disk was outputting text to the printer.

Once you have the text entered, type .END at the beginning of a line to terminate the program. The disk will run, the program will close the file and the READY prompt will appear. Incidentally, speaking of CLOSE, make it a practice to always type and enter this whenever you break a program. Otherwise, you may find that your file is listed in the directory but if you examine the individual programs with DIR (S,I,A), there will be a zero shown for contents!

You can now type NEW to remove the PREP program and load the file name you have selected. Review the text for typos and insure that the proper commands have been given. Occasionally an .IT5 will store as a .IT-5. or if you try to use more than a couple of indents, a margin change command might occur. Once you are satisfied with the text, resave it (filename,A) and then print it as you did with the DEMO program.

At this point you have all the basics of the ELECTRIC SECRETARY program. To become proficient, you must practice processing text. It is a powerful program and you cannot expect to sit down at your TRS-80 and generate perfectly formatted text anymore than you would expect to pick up a guitar and play something by Segovia. You have learned the chords, now you must practice to make the music.

RENUMBERING

Often it will be necessary to add additional text to material previously done. Since the lines advance in steps of 10, you have 9 lines to insert other copy at any point. You can usually get more space by moving a command forward or back 9 numbers.

Occasionally, however, even this will not provide enough space. In this instance, it is necessary to utilize the RENUMBER program. Select this via the MENU. Answer the file name that you want to renumber, then specify 0 and either 100 or 200 for the incremental steps. The program will run and you will see the renumbering on the screen. This file will now be loaded in memory as TEMPNUM. After you have added the text, or otherwise modified it, resave it with the original file name (,A) and KILL "TEMPNUM".

APPENDING

Typical text is usually sufficiently long that it cannot be entered totally at one setting. You may be tired or diverted and type .END before the entry is complete.

The file can later be reloaded under PREP and appended by specifying option "A". This will open a new file called "TEMP" in which you can add to the text. However, the file is vulnerable. Even though the text is recorded in segments, if the disk should reboot to DOS (such as when the oil burner or air conditioner kicks on), the data that was in the TEMP file and the data you are adding will be destroyed because the file was not closed. This can also happen if you fill the disk up on a single disk system.

There is another consideration. Let's say you are preparing a manuscript

that is dozens of pages long. You can keep entering text until the disk is full. However, what happens when you try to edit all this copy? The memory space available is the difference between the DOS requirements (10K) and the maximum, 32K for example. Obviously if you try to edit copy that consumes more than 22K, you will get a OUT OF MEMORY notation when you try to load it. You will not be able to edit the last half of the copy.

COUPLING FILES

The solution is a new command that has not been mentioned called next file. The command is .nx(file name). For example, you have entered four or five pages of single spaced text. You are getting near your memory limit. Before you type .end, first type .nx(file name) on the preceeding line by itself. If, for example, your file name is "sam", then you might type .nxsam1. When you resume typing, you would open a new file called "sam1". After typing another four or five pages of single spaced text, type .nxsam2 and .end. Naturally, when you resume or continune, you would open a file called "sam2" and so on until you are finished or the disk is full. You should be able to get about 12 single spaced pages on a disk, along with DOS and the ELECTRIC SECRETARY PROGRAM, plus the DICTNARY file, which requires 11 grans.

You can also keep the ELECTRIC SECRETARY program on one diskette and the manuscript on a second one. Before you answer the file name request in the program, be sure to insert the manuscript diskette or the program will not be found. You should be able to get 20 single spaced pages on a diskette without the ELECTRIC SECRETARY program. Be sure to leave room for the dictionary file, which requires 11 grans. If you have a multiple disc system, you can store an infinite amount of copy.

By introducing several new commands, you can start to do some very useful word processing. One command is to call a standard paragraph (.st filename). Let's say you are generating a series of form letters. They are almost identical but some require special paragraphs tailored to the particular reader. You can prepare the letter, then at an appropriate point in the text imbed the command .stmoney (for example). If you have also written a file called "money", the program will transfer to this standard paragraph, insert it and then resume typing the letter. You can prepare as many standard paragraph inserts as you wish by giving each one a distinct file name.

Here's a typical problem for word processors. You are sending out letters to your past due customers. They must be reminded in your form letter of how much they owe you and how far past due they are. You can use the wait commmand, but there is a better way. It is called the data file command (.df). In the command area of your letter, insert the .df command without specifying a file name. Then when you get to the part of the letter that says "you owe us...", insert two question marks with spaces on each side. For example, type "You owe us ?? and if you don't pay etc. etc.". When the program finds the double question mark, it will stop before printing the line and display a single question mark on the screen. You would then type in the dollar amount on the TRS-80 keyboard and then press the enter key. The program will then take your data, justify the next line and print it. You can imbed several double question marks in the text for other data such as how far past due, when they are going to be sued and so on.

Let's expand this concept further and really get fancy. All past due letter should have a "nasty paragraph. The degree of "nastiness" depends on how far past due the customer is. You can combine the data file command (.df) and the standard paragraph command (.st) used earlier to completely customize your letters. First, don't forget to put the .df in the command area of the letter. Then, when you reach the place in the text where you want to insert the nasty paragraph, use the command .st??. When the program reaches this point it will stop, but this time you type nasty1, nasty2, nasty3 or whatever standard paragraph you want inserted. Any number of standard paragraphs can be manually inserted in a letter using the .df and .st?? commands.

You can further couple files by imbedding standard paragraph calls within other standard paragraphs. This concept leads to another useful application for the ELECTRIC SECRETARY program called cross coupling.

CROSS COUPLING FILES

Let's say you have a ton of identical form letters to produce. Naturally the ELECTRIC SECRETARY can generate these with ease. But what about the addresses? Do you have to insert the letterhead, type the address and salutation, then run the letter? Certainly not! You can save all this work by cross coupling files with the next file (.nx) and the standard paragraph (.st) commands.

First, open a file called "names", or some such, and repetitively enter the name, address, city/state/zip, a blank line, salutation (Dear Mr. Jones), another blank line then the standard file command (.stletter). Repeat this same

format with the next name until the mailing list is completely entered. Once this file is prepared you never have to do it again, except for updating. Next, prepare a file called "letter". Start with a suitable number of blank lines to miss the letterhead and start the printing at the right line on the page. Then put in the format commands desired (.syl.jul.fl, etc.),the current date, a blank line and then the next file command (.nxnames).

After you have completed the entry of text (and before you type .end), add a new command called repeat file (.rp). This command will close the file, then open it again at the beginning. Don't forget to insert a wait command (.wa) just before the repeat command, so you have time to insert another letterhead.

Note that you can purchase your letterheads printed on rolls. Some types are made for tractor feed and have tear off sprocket margins. If either form of letterhead is used, omit the wait command. With program planning, and the proper use of the ELECTRIC SECRETARY commands, you can start the printing and never touch the computer again until all the letters are printed!

SPECIAL COMMANDS .PK

Poke Command- The poke function of the TRS-80 can be used with the ELECTRIC SECRETARY program. By imbedding the command .pk, you can change the contents of machine language routines. For example, one useful application is an upper case shift lock.

Normally there would be a problem if you want to insert a lot of upper case material when the upper/lower case function has been selected. You would need to hold your little finger on the shift key all the time you are typing the upper case letters.

An easier way is to use the poke command to simply switch back to upper case only. As you may recall from the menu, you can do this by typing &h4019,0 However, poke commands embedded in the ELECTRIC SECRETARY program must be in decimal form. The decimal equivalent of hex 4019 is 16409. Thus, typing the command .pk16409,0
IS THE EQUIVALENT OF A SHIFT LOCK AND UPPER CASE ONLY LETTERS ARE PRINTED. SIMPLY KEEP TYPING ALONG WITHOUT HITTING THE SHIFT KEY UNTIL YOU ARE FINISHED WITH UPPER CASE LETTERS, THEN TYPE .PK16409,1.
and the shift is unlocked. You can also use the poke command .pk16408,1 to echo text on the screen while printing. There are probably many other applications of the poke command that will occur to the user. The poke statements must be on a line by themselves and not embedded in a line of text as was done above.

.SM(n)

* Macro Command- To avoid having to specify the same group of commands over and over, or to shift between formats, these commands can be set up as macro subcommands. The command Set Macro (.sm(n)command, command, command, etc.) establishes as set of commands for a given macro number (n) which are performed by the Execute Macro (.em(n) command. For example, .em7 will perform all the commands set up by command .sm7. You can have up to 10 of these macro groups.

.EM(n)

* Table of Contents- You may want a table of contents printed at the end of a document. This can be accomplished by using the command .co(d). Any data following this command will be printed at the end, along with the page number where the data is to be found.

.OV(n)

* Text Mover- There is another page format command that was not mentioned earlier. You may want to move the entire body of text right or left on the page. This can be done by envoking the over command (.ov(n). The number (n) can be either positive or negative. However, you obviously cannot move the text left beyond the mechanical stop of the printer.

.LFd LF = ✓ RS =
.TPn TP = ✓ goto next page LS =

.PAn PA = Page No NM =
CH = SH
OM =

MARGIN CONTROL

COMMAND	FORMAT
.BM - Bottom Margin ✓	.BMn
.LM - Left Margin ✓	.LMn
.MA - Margin Adjust ✓	.MAn
.OV - Over (Shift text Right) ✓	.OVn
.PL - Page Length ✓	.PLn
.RM - Right Margin ✓	.RMn
.TM - Top Margin ✓	.TMn

FORMATTING

.BL - Blank Line ✓	.BLn
.CE - Center Line ✓	.CEd
.FL - Fill Lines ✓	.FL
.IT - Indent Line ✓	.ITn
.NF - No Fille ✓	.NF
.NJ - No Justify ✓	.NJ
.SP - Spacing ✓	.SPn
.SY - Hyphenate ✓	.SY1
.JU - JUSTIFY ✓	.JU1
END - SAVE FILE	.END

CARRIAGE CONTROL

COMMAND	FORMAT
.HD - Header ✓	.HDd (NO TO KILL NO)
.WA - Wait ✓	.WA
.WT - Wait at Top ✓	.WT1

SPECIAL COMMANDS

.CM - Comment ✓	.CMd
.CO - Talbe Contents ✓	.COd
.DF - Date File ✓	.DFd
.EM - Execute Macro ✓	.EMn
.IN - Index ✓	.IN
.NI - No Index ✓	.NI
.NX - Next File ✓	.NXf
.PK - Poke ✓	.PKn,n
.RP - Repeat File ✓	.RP
.SM - Set Macro ✓	.SMn
.ST - Standard Paragraph ✓	.STf
?? - Input Data	
↑ - Word for Index	

A - ASCII Value of Character in Word to be Hyphenated
A$ - Input Buffer
A9 - Command Indicator
A9$ - Command List
AA - Pointer to Next Overflow Record in Dictnary
AA$ - Area in field statement for Next Overflow Record Pointer in DICTNARY
AB - Number of Words in DICTNARY
AB$ - Area in Field Statement for number of words in DICTNARY
B$ - Data Buffer
B1 - Spacing
B2 - Loop Counter
B9 - Temporary Storage for B1
BM - Bottom Margin
C - Length of C$ During Syllibication
C$ - Output Buffer
CF - Location of Character to be Pulled out of CH$
CG - Loop Counter and Test Variable
CG$ - Temp Change Character Storage
CH - # of Characters in CI$ and CH$
CH$ - Change from String
CI - Temp Location Variable
CI$ - Change to String
CO$() - (150) Contents Array
CO1 - Pointer into CO$
D - Temporary Variable
D$ - Word Being Hyphenated
D1$ - Temporary String
D2$ - Temporary String
DF - Data Input Flag
E - Temporary Variable
E$ - Hyphenation Points for Word (from DICTNARY)
F - File Pointer
F$ - Output String to Add a Word and Its Hyphenation Points into DICTNARY
F1 - Temporary Variable
FL - Fill/No Fill Flag
G - Temporary Variable
G$ - Input String
GG - Standard Header loop counter
GG$ - Temporary storage for A$
GH$ - Temporary storage for B$
GK$ - Temporary storage for C$
H - Loop Counter
H$ - Header
H1 - Page Number
H4 - Temporary storage for I4
I - Loop Counter
I2 - Temporary Variable
I3 - Temporary Variable
I1 - Temporary Variable
I1$ - Tab character
IM - Index Counter
IN - Index Flag
IN$ - Index Sort Array
J - Temporary Variable
J1 - Temporary Variable
J2 - Temporary Variable
K - Number of Characters to be Spread in Justification
K() - (50) Space Locations for Justification
K0 - Forward/Backward Variable
K1 - Temporary Variable

K4 - # of Characters Necessary for Justification of Line
K5 - Length of C$
K7 - Forward/Backward Flag
KD - Temporary Variable
KE - # of Spaces (or Increments) to be Added Each Time
KF - # of Times to use KE Before Adding K0 to It.
KG - Temporary Variable
KQ - Position to break A$
L$ - Temporary Storage for Left Side of C$
L1 - Right Margin
L2 - Left Margin
L3 - Line Length
L4 - Indent
L5 - Print Variable for Indent
L8 - Comma position in Left Side string
L9 - Line Count
LM() - Left margin array
LS() - (100) Left Side array
M - Loop counter for DICTNARY creation
MA$() - (10) Macro array
MB - Temporary storage for B1
MD - Temporary storage for KD
ME - Temporary storage for KE
MF - Temporary storage for KF
N - Sort Variable
N1 - Drive Number for File
N1$ - File Name
N2$() - (15) File Names in Use
NJ - Justify Flag
NJ() - (10) Last in - first out (LIFO) stack for NJ
O1 - Temporary storage for L1
O2 - Temporary storage for L2
O3 - Temporary storage for L3
O4 - Temporary storage for L4
OP - Output Device
OU - Flag for Standard Header
OV - Spaces to left side of page
P - Page Length
P1 - Pointer into DICTNARY
P9 - Change Variable for Page #
PP1 - Line # for File Output
Q - Temporary Variable
Q1$ - Data File Name ("" if CRT)
R8 - Comma position in Right Side string
R$ - Data Field in DICTNARY File
RM() - Right margin array
RR$ - Overflow pointer in DICTNARY record
RS() - (100) Right Side array
SH$() - (10) Standard Header array
SY - Syllabication Flag
T1 - Hold Area for L1
T2 - Hold Area for L2
T3 - Margin Adjust Amount
TM - Top Margin
WT - Wait Flag
U1 - Stack pointer for NJ storage
X1-X9 - Environment preservation storage
Z - Temporary Variable
Z$ - Line Number for Line being Processed
Z9 - Temporary Variable
ZZ$ - Dummy Input Variable for Wait

ELECPR

ELECTRIC SECT - PRINT

```
10 REM FORMATTED FOR 10 PITCH TYPE
20 CLS:CLEAR10000
30 PRINTSTRING$(63,"*").PRINTTAB(15)"THE ELECTRIC SECRETARY- PRINTING":PRINTSTRING$(63,"*")
40 DEFINT A-Z :DIM CO$(150),N2$(15),K(50),LS(100),RS(100)
50 LINEINPUT"FILENAME  ";N1$
60 ONERRORGOTO70:Q=5:GOSUB2620:GOTO90
70 IFERR/2+1=54 THEN RESUME80
80 ON ERRORGOTO0:PRINT"FILENAME  <";N1$;"> NOT FOUND": FORA=1TO1000:NEXT:RUN
90 REM
120  OF=1:PRINT"PRINTER"
160 L1=70:T1=70:L2=0:T2=0:L3=L1-L2:H1=1:P=66:IN=0:IM=0:CO1=0
170 DF=0:L9=1:B1=0:BM=6:TM=0:F=5:NJ=1:PP1=10
180 A9$=".II. IT. PL. RP. DF. MA. WT. LM. RM. HD. CE. NF. FL. BM. TM. JU. NJ. SP. LF. TP"
190 A9$=A9$+". ST. PK. WA. BL. PA. CO. IN. NI. NX. CH. SY. RS. LS. SM. EM. NM. SH. OM. CM. OV"
200 A$="":B$="":C$="":H$=""


210 '****** PROCESS A LINE *******
220 '** GET NEXT LINE
230 IF EOF(F)THEN IF OU THEN F=F-1:RETURN ELSE GOSUB 1920:
     IFF<6THEN2310ELSECLOSEF.N2$(F)="":F=F-1:GOTO230
240 LINEINPUT#F,A$:IF A$=""THEN 230
250 Z=INSTR(A$,"'"):IF Z THEN Z$=LEFT$(A$,Z-1):A$=MID$(A$,Z+1)ELSEPRINT"NO ' IN "A$:Z$=LEFT$(A$,5)
340 '** INPUT FROM DATA FILE OR CRT
350 IF DF=0 THEN 420
360 D=INSTR(A$,"??").IF D=0 THEN 420
370 IF N2$(3)=""THEN LINEINPUT"?";G$:GOTO 400
380 IF EOF(3)THEN IF F=5THEN PRINT "OUT OF DATA":CLOSE:ENDELSE CLOSE3: DF=0:CLOSEF:F=F-1:GOTO230
390 LINEINPUT#3,G$:G$=MID$(G$,INSTR(G$,"?")+1)
400 IFD=1THENA$=G$+MID$(A$,D+2):GOTO350 ELSE A$=LEFT$(A$,D-1)+G$+MID$(A$,D+2):GOTO350
410 '** CHECK FOR COMMAND
420 IF LEFT$(A$,1)=". "THEN 1290
430 '** PRINT A$ IF NOT FILLING
440 IF FL THEN C$=A$:GOSUB 1970:GOTO230
450 '** ADD A$ TO B$
460 IF LEN(B$)+LEN(A$)>255THEN K0=240-LEN(B$):B$=B$+LEFT$(A$,K0):A$=MID$(A$,K0+1):GOTO500
470 IF RIGHT$(B$,1)<>" "THEN B$=B$+" "
480 B$=B$+A$:A$=""
490 '** FILL C$ FROM B$
500 IF LEFT$(B$,1)=" "THEN B$=MID$(B$,2):GOTO500
510 IF LEN(B$)<L3THEN IF LEN(A$) THEN 480 ELSE 230
520 GOSUB 740:IF L3=LEN(C$) THEN GOSUB 2680.NJ=0:GOSUB 1970.GOSUB2690:GOTO510
530 '** JUSTIFY C$
540 K=1:K1=1
560 IF L4<0 THEN L3=L3+L4:L$=LEFT$(C$,ABS(L4)):C$=MID$(C$,ABS(L4)+1)
570 K4=L3-LEN(C$)
580 ONNJGOSUB610
590 IF L4<0 THEN L3=L3-L4:C$=L$+C$
600 GOSUB 1970:GOTO 510

610 '**** SUB: ADD BLANKS FROM L TO R
620 K=-1
630 IF K1 THEN K=K+1:K1=INSTR(K1+1,C$," "):K(K)=K1:GOTO630
640 KE=K4/K:KF=K4-KE*K
650 IF K7=0 OR K=1 OR K=K4 OR KF<1THEN K7=1:K0=1:KF=K-KFELSE K7=0:K0=-1:KE=KE+1
660 E$="":K1=1
670 FOR I=0 TO K-1:E$=E$+MID$(C$,K1,K(I)-K1)+STRING$(KE," "):KF=KF-1:K1=K(I):IF KF=0 THEN KE=KE+K0
680 NEXT:C$=E$+MID$(C$,K(K-1))
690 RETURN
```

```
 730 '** SUB FILL C$
 740 IF L3>LEN(B$) THEN C$=B$:B$="":RETURN
 750 FOR I=L3+1 TO 2 STEP-1
 760  IF MID$(B$,I,1)=" "THEN IF MID$(B$,I-1,1)<>" " THEN 800
 770  IF MID$(B$,I,1)="-"THEN IF I<=L3THEN C$=LEFT$(B$,I):B$=MID$(B$,I+1):RETURN
780 NEXT
 790 PRINT"ERROR: LINE TOO LONG BETWEEN MARGINS. ";L3;"SPACES BETWEEN MARGINS":
PRINT"LINE TO BE SPLIT IS.":PRINTZ$:PRINTB$:STOP
 800 C$=LEFT$(B$,I-1):B$=MID$(B$,I+1)
 810 IF MID$(B$,1,1)=" " THEN B$=MID$(B$,2):GOTO810
 820 C=LEN(C$):IF L3-C<6 OR SY=0 THEN RETURN

 830 '** SYLLABICATION
 840 D=INSTR(B$," "):IF D THEN D$=LEFT$(B$,D-1)ELSE D$=B$
 850 IF RIGHT$(D$,1)="." THEN RETURN
 860 D=LEN(D$):IF D<9 THEN RETURN
870 D1$=""
 880 P1=0:FOR I=1 TO D:A=ASC(MID$(D$,I))
 890    IF A<65 OR 95<A THEN 920
 900    D1$=D1$+CHR$(A):P1=P1+A
910 NEXT
920 GOSUB 1080
 930 D=LEN(D1$):IF D<9 THEN RETURN
 940 GET 4,P1: E=INSTR(R$,D1$)
 950 IFE=0THENIFINSTR(R$,CHR$(126))>(238-D)THENP1=P1+19:
       GOSUB1080:GOTO920ELSE990
 960 F1=E+D+1:E$=MID$(R$,F1,INSTR(F1,R$,CHR$(125))-F1)
 970 FORI=LEN(E$)TO1STEP-1:G=ASC(MID$(E$,I)):
       IFC+G<L3THENC$=C$+" "+LEFT$(B$,G-1)+"-":B$=MID$(B$,G):
       RETURNELSENEXT
980 RETURN
 990 PRINT:D2$="--> "+D1$+" <--":
       PRINTD2$;"    IS NOT IN MY HYPHENATION DICTIONARY"
1000 PRINT"PLEASE TYPE IT IN WITH DASHES BETWEEN SYLLABLES."
1010 G$="":LINEINPUT"?";G$:IFG$=""THEN RETURN
1020 E$="":J=0:G=1
1030 H=INSTR(G,G$,"-"):IFHTHENE$=E$+CHR$(H-J):J=J+1:G=H+1:GOTO1030
1040 F$=D1$+CHR$(124)+E$+CHR$(125)+CHR$(126)
1050 G$=LEFT$(R$,INSTR(R$,CHR$(126))-1)
1060 IF LEN(G$)+LEN(F$)<244
        THENLSETR$=G$+F$:PUT4,P1:
        GET4,1:AB=AB+1:LSETAB$=MKI$(AB):PUT4,1:
        GOTO970
1070 P1=P1+19:GOSUB980:GOTO1060
1080 P1=(INT(P1)-51*FIX(INT(P1)/51))+1 : RETURN

1090 '******* SUB:MAKE CHARACTER SUBSTITUTIONS
1100 RETURN
```

```
1150 '********** SUB: HOF & HEADER
1160 IF L9=TM+1
        THEN 1180
        ELSE FOR L9=L9 TO P+TM:
        GOSUB 2080:
        NEXT:H1=H1+1
1170 IF WT THEN INPUT"WAIT -- ALIGN PAPER";ZZ$
1180 Z9=B1:GOSUB2680:L9=TM+1:IF H$="NO" THEN 1230
1190 IF H$=""THEN C$="" ELSE C$="- "
1200 C$=H$+C$+" PAGE"+STR$(H1)
1210 IF B1<1THENB1=1
1220 H4=L4:NJ=0:L4=0:GOSUB1980:L4=H4
1230 GG$=A$:X1=FL:X2=SY:X3=LM:LM=0:X4=T1:X5=T2:X6=KD:X7=KE:X8=KF
1240 OU=1:FORGG=0TO9:IFLEN(SH$(GG))THENA$=SH$(GG):GOSUB1720
1250 NEXT:OU=0
1260 A$=GG$:FL=X1:SY=X2:LM=X3:T1=X4:T2=X5:KD=X6:KE=X7:
        KF=X8:B1=Z9:GOSUB2690
1270 RETURN


1280 '******** SUB: COMMAND HANDLER  **********
1290 K7=0
1300 GOSUB1920:AC=ASC(MID$(A$,2)):IFAC>96THENMID$(A$,2,1)=
        CHR$(AC-32):MID$(A$,3,1)=CHR$(ASC(MID$(A$,3))-32)
1310 AC=ASC(MID$(A$,2)):IFAC>96THENMID$(A$,2,1)=CHR$(AC-32):
        MID$(A$,3,1)=CHR$(ASC(MID$(A$,3))-32)
1320 A9=INSTR(A9$,MID$(A$,1,3))/3
1330 ONA9GOTO1410,1440,1450,1470,1490,1510,1520,1530,1540,1560,
        1570,1580,1610,1620,1630,1640,1680,1690,1700,1720,1730,
        1740,1750,1770,1790,1800,1810,1820,1830,1860,1380,1350,
        1590,1600,1650,1660,1670,1420,1550
1340 PRINTMID$(A$,1,3)," IS A BAD COMMAND ON LINE ";Z$;:GOTO1890


1350 LM=0:L8=0
1360 L8=INSTR(L8+1,A$,","):IFL8 THEN LS(LM)=VAL(MID$(A$,L8+1)):
        LM=LM+1:GOTO1360
1370 LS(LM)=0:L8=0:GOTO1520
1380 RM=0:R8=0
1390 R8=INSTR(R8+1,A$,","):IFR8 THEN RS(RM)=VAL(MID$(A$,R8+1)):
        RM=RM+1:GOTO1390
1400 RS(RM)=0:R8=0:GOTO1530
1410 L4=VAL(MID$(A$,4,1)):L3=L3-L4:GOTO1890
1420 IFMID$(A$,4,1)="?"THENPRINTMID$(A$,5)
1430 GOTO1890
1440 P=VAL(MID$(A$,4)):GOTO1890
1450 CLOSE F:Q=F:N1$=N2$(F)
1460 GOSUB2620:GOTO1890
1470 N1$=MID$(A$,4):DF=1:
        IF (LEN(N1$))AND(N1$<>N2$(3))
         THENQ=3:GOSUB2620
1480 N2$(3)=N1$:GOTO1890
1490 T3=VAL(MID$(A$,4)):IFT3THEN L1=L1-T3:L2=L2+T3
         ELSE L1=T1:L2=T2
1500 L3=L1-L2:GOTO1890
```

```
 1510 WT=VAL(MID$(A$,4)):GOTO1890
 1520 L2=VAL(MID$(A$,4)):T2=L2:L3=L1-L2:GOTO1890
 1530 L1=VAL(MID$(A$,4)):T1=L1:L3=L1-L2:GOTO1890
 1540 H$=MID$(A$,4):GOTO1900
 1550 OV=VAL(MID$(A$,4)):GOTO1890
 1560 C$=STRING$(INT((L3-LEN(MID$(A$,4)))/2)," ")+MID$(A$,4):
        GOSUB2680:NJ=0:GOSUB1970:GOSUB2690:GOTO1900
1570 FL=1:GOTO1890
1580 FL=0:GOTO1890
 1590 I=VAL(MID$(A$,4)):MA$(I)=MID$(A$,5):GOTO1900
 1600 I=VAL(MID$(A$,4)):A$=LEFT$(A$,4)+MA$(I)+MID$(A$,5):GOTO1890

 1610 BM=VAL(MID$(A$,4)):GOTO1890
 1620 TM=VAL(MID$(A$,4)):GOTO1890
 1630 NJ=VAL(MID$(A$,4)):GOTO1890
1640 NJ=0:GOTO1890
 1650 O3=L3:O4=L4:O2=L2:O1=L1:L2=VAL(MID$(A$,4)):
        L1=VAL(MID$(A$,INSTR(A$,",")+1)):L3=L1-L2:L4=0:
        GOTO1890
 1660 I=VAL(MID$(A$,4)):SH$(I)=".ST"+MID$(A$,5):GOTO1890
 1670 L3=O3:L4=O4:L2=O2:L1=O1:GOTO1890
 1680 B1=VAL(MID$(A$,4))-1:GOTO1890
 1690 C$=MID$(A$,4):K=1:K1=1:K4=L3-LEN(C$):GOSUB2680:NJ=0:
        GOSUB620:GOSUB1970:GOSUB2690:GOTO230
 1700 IF L9>P-BM-VAL(MID$(A$,4))
        THEN GOSUB 1160
1710 GOTO1890

 1720 F=F+1:N1$=MID$(A$,4):Q=F:GOSUB2620:GOTO1890
 1730 POKEVAL(MID$(A$,4)),VAL(MID$(A$,INSTR(A$,",")+1)):GOTO1890
 1740 PRINT"WAIT ";MID$(A$,4);:LINEINPUT" ";Z2$:GOTO1890
 1750 B9=B1:B1=VAL(MID$(A$,4)):IF B1=0 THENB1=B9:GOTO1890
1760 GOSUB2010:B1=B9:GOTO1890
 1770 P9=VAL(MID$(A$,4)).IF P9 THEN H1=P9-1
1780 GOSUB1160:GOTO1890
 1790 CO$(CO1)=MID$(A$,4)+CHR$(127)+STR$(H1):CO1=CO1+1:GOTO1890
 1800 IN=1:OPEN"O",1,"INDEX":GOTO1890
 1810 IN=0:GOTO1890
 1820 N1$=MID$(A$,4):Q=F:GOSUB2620:GOTO1900

 1830 GOTO1890:REM .CH
 1860 SY=VAL(MID$(A$,4)):CLOSE4:IF SY=0 THEN1890
 1870 OPEN"R",4,"DICTNARY":FIELD#4,245 AS R$,2 AS AB$,8 AS G$:
       GET 4,1
 1880 IF G$="DICTNARY"THEN AB=CVI(AB$)
      ELSELSETG$="DICTNARY":LSETR$=CHR$(125)+CHR$(126):
       LSETAB$=MKI$(0):
          FORM=1TO51:PUT4,M:NEXTM:GOTO1890

 1890 A9=INSTR(2,A$,". "):IFA9<>0THENA$=MID$(A$,A9):GOTO1310
1900 B$="".GOTO230
```

```
1910 '**** SUB: RUN OUT B$
1920 IF B$=""THEN RETURN
1930 C$=B$:B$="":GOSUB2680:NJ=0
1940 IF(NF=0)AND(L9+B1<P)THENGOSUB1980ELSEGOSUB1970
1950 GOSUB2690:RETURN

1960 '***** SUB. PRINT A LINE *****
1970 IF L9>P-BM THEN GH$=B$:GK$=C$:GOSUB1160:B$=GH$:C$=GK$
1980 GOSUB1100
1990 GOSUB2140:IFLM=0 THEN L3=L3+L4
2000 L4=0:L9=L9+1:IFL9=0THENLPRINT""
2010 IF B1=0 THEN GOTO2060
2020 FOR B2=1 TO B1
2030  GOSUB 2080
2040  L9=L9+1
2050 NEXT
2060 C$="":RETURN
2070 '***** SUB: OUTPUT BLANK LINE TO CORRECT DEVICE
2080 REM
2100 LPRINT" ":RETURN
2130 '**** SUB: SEND OUTPUT TO CORRECT DEVICE
2140 REM
2170 LPRINTTAB(L2+L4+OV);:IFL4>0THENL5=1ELSEL5=ABS(L4)+1
2180 LPRINT LEFT$(C$,L5);
2200 LPRINTMID$(C$,L5+1):RETURN

2290 '***** FINAL OUTPUT **********
2300 '*** CLOSE FILES
2310 REM
2600 GOSUB1160:RUN
2610 '***** SUB: OPEN FILES
2620 CLOSE#0.N2$(0)=N1$
2630 IF RIGHT$(N1$,1)=" "THENN1$=LEFT$(N1$,LEN(N1$)-1):GOTO2630
2640 IF LEFT$(RIGHT$(N1$,2),1)=":"THEN N1=VAL(RIGHT$(N1$,1))ELSE N1=0
2650 OPEN"I",0,N1$:LINEINPUT#0,G$
2660 IFG$<>"0 '" THENPRINTN1$;"  IS NOT AN ASCII FILE";:FORZ=1TO1000:NEXT:RUNELSE RETURN
2680 U1=U1+1:NJ(U1)=NJ:RETURN
2690 NJ=NJ(U1):U1=U1-1:RETURN
```

```
.syl
.jul
20 March, 1979
.bll
Mr. John Jones
333 Hidden Vale Rd.
Jumpoff, California 93789
.bll
Dear Mr. Jones:
.bll.fl.it5
It has come to our attention that your account of $1500.00
is seriously past due.
.bll.it5
In order to protect your credit rating with us, it will be
necessary for you to bring this account current by the end
of the week.
.bll.it5
If this is not possible, I would like to hear from you at
your earliest convenience in order that we may discuss
the manner in which this account can be cleared.
.bll.it5
Please call me at 567-8901 between 8am and 5:30pm.
.nf
.bll
Sincerely,
.bl3
L. A. Stoner
Credit Manager
.bl2
LAS/
```

Sample of Letter to be Entered For Demonstration

DEMO

```
0 '
10 '.CM PUT HYPHENATE COMMAND HERE
20 '.LM5
30 '.RM60
40 '.PL50
50 '.JU1
60 '.SP1
70 '.NF
80 'THE ELECTRIC SECRETARY
90 'BY
100 'THE PERIPHERAL PEOPLE
110 '.BL
120 '.BL1
130 'OPERATING INSTRUCTIONS
140 '.BL1
150 '.CM START OF TEXT
160 'INTRODUCTION
170 '.BL1
180 '.IT5
190 '.CM PLACE FILL COMMAND HERE
200 'THE ELECTRIC SECRETARY PROGRAM WAS WRITTEN TO ADAPT BASIC TRS-80 BUSINESS SYSTEMS
210 '(COMPUTER, SCREEN, DISK AND PRINTER) TO WORD PROCESSING APPLICATIONS.
220 'IT REQUIRES A MINIMUM MEMORY OF 20K (INCLUDING DOS) AND THEREFORE IS USED ON 32 AND 48K SYSTE
230 'HOWEVER, WITHOUT MODIFICATION, THIS CONFIGURATION WILL PRINT UPPER CASE LETTERS ONLY.
240 '.IT5
250 'WHILE THIS IS SATISFACTORY FOR MANY COMMERCIAL APPLICATIONS,
260 'THE TRS-80 CAN BE EASILY MODIFIED FOR UPPER/LOWER CASE PRINTING AND VIDEO DISPLAY.
270 'WITH THIS MODIFICATION, PLUS THE ADDITION OF AN RS-232 ADAPTER AND
280 'A DAISY WHEEL PRINTER (QUME, DIABLO, MULTI-TERM, ETC.)
290 ', THE "LOWLY" TRS-80 WILL OUTPERFORM AN IBM "MAG-CARD II" $12,000 WORD PROCESSING SYSTEM,
300 'IN ADDITION TO ITS PRIMARY FUNCTION AS A COMPUTER.
310 '.BL1
320 'ADVANTAGES
330 '.BL1
340 '.IT5
350 'THE ELECTRIC SECRETARY IS A DISK BASED WORD PROCESSING PROGRAM WRITTEN IN BASIC SPECIFICALLY
360 'WRITING THE PROGRAM IN BASIC OFFERS NUMEROUS ADVANTAGES OVER MACHINE LANGUAGE WORD PROCESSING
370 'THE OPERATOR CAN MORE NEARLY UNDERSTAND HOW THE PROGRAM WORKS, BY LISTING IT.
380 'THE OPERATOR ALSO HAS A DEGREE OF CONTROL THAT IS NOT AVAILABLE WITH MACHINE LANGUAGE PROGRAM
390 'FOR EXAMPLE, DEFAULT VALUES FOR MARGINS, PAGE LENGTH AND SO ON CAN
400 'BE MODIFIED BY THE OPERATOR FOR MOST OFTEN USED PAGE FORMATS.
410 '.IT5
420 'THE ELECTRIC SECRETARY INCORPORATES A HYPHENATING DICTIONARY THAT IS NOT FOUND IN MACHINE LAN
430 'IN TYPICAL TEXT, SOONER OR LATER, A WORD WILL COME ALONG THAT MUST BE HYPHENATED.
440 'LESSER PROGRAMS SIMPLY LEAVE A GAP AT THE END OF THE LINE WHEN THIS OCCURS
450 'OR ELSE LEAVE BIG GAPS BETWEEN WORDS TO FILL OUT THE LINE.
460 'THIS DESTROYS THE SYMMETRY OF A BEAUTIFULLY FORMATTED PAGE OF TEXT.
470 '.IT5
480 'ANOTHER ADVANTAGE OF THE ELECTRIC SECRETARY IS THE USE OF DISK STORAGE RATHER THAN CASSETTES.
490 'ONCE THE TEXT IS COMPLETELY ENTERED, IT IS SAVED ON DISK AS A BASIC LINE NUMBERED PROGRAM.
500 'THIS FILE CAN THEN BE LOADED FROM THE DISK INTO MEMORY AND TOTALLY MANIPULATED USING THE TRS-
510 'WORDS OR SENTENCES CAN BE ADDED, DELETED, TYPOS CORRECTED AND ON.
520 'ONCE THE OPERATOR IS SATISFIED WITH THE TEXT, IT CAN BE RESAVED ON DISK FOR LATER USE.
```

MARDOT MANAGEMENT COMPANY
77 HILLSIDE AVE
WILLISTON PARK NY 11596

MOHEAD

SEPTEMBER 20, 1979

GENTLEMEM

IN ORDER TO COMPLY WITH VARIOUS GOVERNMENTAL REGULATIONS IT IS NECESSARY THAT WE HAVE A RECORD OF YOUR FEDERAL EMPLOYER IDENTIFICATION NUMBER.

WE WOULD APPRECIATE IT IF YOU WOULD FILL IN YOUR NUMBER IN THE SPACE BELOW AND RETURN THIS FORM IN THE ENCLOSED ENVELOPE.

VERY TRULY YOURS

MARDOT MANAGEMENT CO.

TENANT NAME : COLONIAL PRESS LTD.

FEDERAL I.D. NO. : _____

MDHEAD - FILE

```
0 '
10 '.CM MARDOT LETTERHEAD - MDHEAD
20 '.BL1.NF.MONO
30 '.SY1.RM80
35 '.WA
40 '.LM25
50 'MARDOT MANAGEMENT COMPANY
60 '.LM30
70 '77 HILLSIDE AVE
80 '.LM26
90 'WILLISTON PARK NY 11596
100 '.JU1.DF
110 '.LM55.BL3
120 'SEPTEMBER 20, 1979
130 '.LM0.BL3
140 'GENTLEMEM
150 '.FL.BL1
160 '.IT5
170 'IN ORDER TO COMPLY WITH VARIOUS GOVERNMENTAL REGULATIONS
180 'IT IS NECESSARY THAT WE HAVE A RECORD OF YOUR FEDERAL EMPLOYER
190 'IDENTIFICATION NUMBER.
200 '.BL1
210 '.IT5
220 'WE WOULD APPRECIATE IT IF YOU WOULD FILL IN YOUR NUMBER IN THE SPACE BELOW AND RETURN THIS FO
230 '.BL1
240 '.LM45
250 'VERY TRULY YOURS
260 '.BL2
270 '.LM43
280 'MARDOT MANAGEMENT CO.
290 '.BL10
300 '.LM10
310 'TENANT NAME       : ??
320 '.BL1
330 'FEDERAL I.D. NO.  :
340 '.LM29
350 '_____
360 '.RP
```

CLIENT

WALTER J SMITH
FINANCIAL CONSULTANT
77 HILLSIDE AVE
WILLISTON PARK NY 11596

516-334-8387

< NAME >
< ADD1 >
< ADD2 >

****

STATEMENT OF ACCOUNT < AS OF DATE >

BALANCE FORWARD < DATE >	$ XXX.XX
LESS PAID ON ACCOUNT	$ XXX.XX
ACOUNTING SERVICES FOR < PERDID >	$ XXX.XX
TOTAL DUE	$ XXX.XX

CL181LL

```
0 '
10 '.BL1.NF.SY1.HDNO
20 '.LM24.IT3
30 'WALTER J SMITH
40 'FINANCIAL CONSULTANT
50 '.LM22.IT5
60 '77 HILLSIDE AVE
70 'WILLISTON PARK NY 11596
80 '.BL1.LM27
90 '516-334-8387
100 '.JU1.DF
110 '.BL1.LM0
120 '??
130 '??
140 '??
150 '.BL3
160 '*****************************************************************************************************
170 '.BL2
180 'STATEMENT OF ACCOUNT                                    ??
190 '.BL1.LM10
200 'BALANCE FORWARD ??                    $ ??
210 '.BL1
220 'LESS PAID ON ACCOUNT                        $ ??
230 '.BL1.LM5
240 'ACOUNTING SERVICES FOR ??               $ ??
250 '.BL1.LM10
260 'TOTAL DUE                                   $ ??
270 '.BL39
280 '.RP
```

MARDOT MANAGEMENT COMPANY
77 HILLSIDE AVE
WILLISTON PARK NY 11596

< DATE >

< NAME >
< ADD1 >
< ADD2 >

DEAR TENANT

RE: BASE RENT ADJUSTMENT

PURSUANT TO THE TERMS OF PARAGRAPH < PAR. NO. > OF YOUR LEASE, YOUR MONTHLY RENTAL HAS BEEN ADJUSTED IN ACCORDANCE WITH THE APPLICABLE CONSUMER PRICE INDEX FOR THE SUCCESSIVE SIX-MONTH ANNIVERSARY OCCURING ON < DATE > .

YOUR NEW MONTHLY RENTAL FOR THE PERIOD < DATE > THROUGH < DATE > IS $ < 000.00 > . WHEN FORWARDING YOUR < MONTH > RENT OF $ < 000.00 > , PLEASE INCLUDE AN ADDITIONAL $ < 00.00 > REPRESENTING THE RETROACTIVE AMOUNTS DUE FOR THE MONTHS OF < MONTH AND MONTH > .

VERY TRULY YOURS,

MARDOT MANAGEMENT COMPANY

```
0 '
10 '.CM MARDOT CPI ADJ TENANT LETTER - CPIADJ
20 '.BL1.NF.HDNO
30 '.SY1
40 '.LM25,
50 'MARDOT MANAGEMENT COMPANY
60 '.LM30
70 '77 HILLSIDE AVE
80 '.LM26
90 'WILLISTON PARK NY 11596
100 '.JU1.DF
110 '.LM55.BL3
120 '??
130 '.LM0.BL3
140 '??
150 '??
160 '??
170 '.BL1
180 'DEAR TENANT
190 '.BL1
200 '.LM25
210 'RE: BASE RENT ADJUSTMENT
220 '.BL1.FL IT5
230 '.LM0
240 'PURSUANT TO THE TERMS OF PARAGRAPH ??
250 'OF YOUR LEASE, YOUR MONTHLY RENTAL HAS BEEN ADJUSTED IN
260 'ACCORDANCE WITH THE APPLICABLE CONSUMER PRICE INDEX
270 'FOR THE SUCCESSIVE SIX-MONTH ANNIVERSARY OCCURING ON ?? .
280 '.BL1.IT5
290 'YOUR NEW MONTHLY RENTAL FOR THE PERIOD ?? THROUGH ?? IS $ ?? .
300 'WHEN FORWARDING YOUR ?? RENT OF $ ?? ,
310 'PLEASE INCLUDE AN ADDITIONAL $ ??
320 'REPRESENTING THE RETROACTIVE AMOUNTS DUE FOR THE MONTHS
330 'OF ?? .
340 '.NF
350 '.BL1
360 '.LM30
370 'VERY TRULY YOURS,
380 '.BL3
390 '.LM28
400 'MARDOT MANAGEMENT COMPANY
410 '.BL2
```

```
2380 P1=0:
       FOR I=1 TO D:
       A=ASC(MID$(D$,I)):A=A+32*(95<A) ^        ????


5540 CF=INSTR(CH$,CG$):IF CF=1
     THEN CH$=MID$(CH$,2):CI$=MID$(CI$,2)
     ELSE IF CF=CH AND CH<>0
          THEN CH$=LEFT$(CH$,CF-1):CI$=LEFT$(CI$,CF-1)
          ELSE CH$=LEFT$(CH$,CF-1)+MID$(CH$,CF+1):
               CI$=LEFT$(CI$,CF-1)+MID$(CI$,CF+1$@,K_ZZN_"1@" @!MMGD!%JMGDMM!C^^*
```

ELEC PR

```
10 REM FORMATTED FOR 10 PITCH TYPE
20 CLS:CLEAR7000:CLEAR FRE(0)
30 PRINTSTRING$(63,"*"):PRINTTAB(15)"THE ELECTRIC SECRETARY- PRINTING":PRINTSTRING$(63,"*")
40 DEFINT A-Z :DIM CO$(150),N2$(15),K(50),LS(100),RS(100)
50 LINEINPUT"FILENAME  ";N1$
60 ONERRORGOTO70:Q=5:GOSUB2620:GOTO90
70 IFERR/2+1=54 THEN RESUME80
80 ON ERRORGOTO0:PRINT"FILENAME  <";N1$;"> NOT FOUND": FORA=1TO1000:NEXT:RUN
90 REM
100 A$="":INPUT"CRT PRINTER OR FILE (C,P OR F) ";A$
110 IF A$="" THEN 50
120 IF LEFT$(A$,1)="P"THEN OP=1:PRINT"PRINTER":GOTO160
130 REM IF LEFT$(A$,1)<>"F"THENOP=3:PRINT"CRT":GOTO370
140 REM OP=2:N1$="":LINEINPUT"FILENAME";N1$:IF N1$=""THEN280
150 REM Q=2
160 L1=70:T1=70:L2=0:T2=0:L3=L1-L2:H1=1:P=66:IN=0:IM=0:CO1=0
170 DF=0:L9=1:B1=0:BM=6:TM=0:F=5:NJ=1:PP1=10:II$=CHR$(9)
180 A9$=":[[/IT. PL. RP. DF. MA. WT. LM. RM. HD. CE. NF. FL. BM. TM. JU. NJ. SP. LF. TP"
190 A9$=A9$+". ST. PK. WA. BL. PA. CO. IN. NI. NX. CH. SY. RS. LS. SM. EM. NM. SH. OM. CM. OV"
200 A$="":B$="":C$="":H$=""
210 '****** PROCESS A LINE *******
220 '** GET NEXT LINE
230 IF EOF(F)THEN IF OU THEN F=F-1:RETURN ELSE GOSUB 1920:IF F<6 THEN 2310 ELSE CLOSE F:N2$(F)="":F=F
240 LINEINPUT#F,A$:IF A$=""THEN 230
250 Z=INSTR(A$,"'"):IF Z THEN Z$=LEFT$(A$,Z-1):A$=MID$(A$,Z+1)ELSEPRINT"NO ' IN "A$:Z$=LEFT$(A$,5)
260 '** CHANGE TABS TO BLANKS
270 II=INSTR(A$,II$):IF II THEN MID$(A$,II,1)=" ":A$=LEFT$(A$,II)+STRING$(7-(II-1)," ")+"MID$(A$,II+
280 '** ADD WORDS TO INDEX
290 IF IN=0 THEN 350
300 I2=INSTR(A$,"["):IF I2=0THEN350
310 I3=INSTR(I2,A$," "):IF I3=0 THEN I3=LEN(A$)
320 PRINT#1,MID$(A$,I2+1,I3-I2)+CHR$(127)+STR$(H1):IF I2>1 THEN A$=LEFT$(A$,I2-1)+MID$(A$,I2+1)ELSE A
330 IM=IM+1:GOTO 300
340 '** INPUT FROM DATA FILE OR CRT
350 IF DF=0 THEN 420
360 D=INSTR(A$,"??"):IF D=0 THEN 420  else D1=D+2
370 IF N2$(3)=""THEN LINEINPUT"?";G$:GOTO 400
380 IF EOF(3)THEN IF F=5THEN PRINT "OUT OF DATA":CLOSE:ENDELSE CLOSE3: DF=0:CLOSEF:F=F-1:GOTO230
390 LINEINPUT#3,G$:G$=MID$(G$,INSTR(G$,"'")+1)
400 IFD=1THENA$=G$+MID$(A$,D+2):GOTO350 ELSE A$=LEFT$(A$,D-1)+G$+MID$(A$,D+2):GOTO350
410 '** CHECK FOR COMMAND
420 IF LEFT$(A$,1)="."THEN 1290
430 '** PRINT A$ IIF NOT FILLING
440 IF FL THEN C$=A$:GOSUB 1970:GOTO230
450 '** ADD A$ TO B$
460 IF LEN(B$)+LEN(A$)>255THEN KQ=240-LEN(B$):B$=B$+LEFT$(A$,KQ):A$=MID$(A$,KQ+1):GOTO500
470 IF RIGHT$(B$,1)<>" "THEN B$=B$+" "
480 B$=B$+A$:A$=""
```

```
490 '** FILL C$ FROM B$
500 IF LEFT$(B$,1)=" "THEN B$=MID$(B$,2):GOTO500
510 IF LEN(B$)<L3THEN IF LEN(A$) THEN 480 ELSE 230
520 GOSUB 740:IF L3=LEN(C$) THEN GOSUB 2680:NJ=0:GOSUB 1970:GOSUB2690:GOTO510
530 '** JUSTIFY C$
540 K=1:K1=1
550 IF OP>1 THEN IF NJ>1 THEN NJ=1
560 IF L4<0 THEN L3=L3+L4:L$=LEFT$(C$,ABS(L4)):C$=MID$(C$,ABS(L4)+1)
570 K4=L3-LEN(C$)
580 ON NJ GOSUB 610,700,710
590 IF L4<0 THEN L3=L3-L4:C$=L$+C$
600 GOSUB 1970:GOTO 510
610 '**** SUB: ADD BLANKS FROM L TO R
620 K=-1
630 IF K1 THEN K=K+1:K1=INSTR(K1+1,C$," "):K(K)=K1:GOTO630
640 KE=K4/K:KF=K4-KE*K
650 IF K7=0 OR K=1 OR K=K4 OR KF<1THEN K7=1:K0=1:KF=K-KFELSE K7=0:K0=-1:KE=KE+1
660 E$="":K1=1
670 FOR I=0 TO K-1:E$=E$+MID$(C$,K1,K(I)-K1)+STRING$(KE," "):KF=KF-1:K1=K(I):IF KF=0 THEN KE=KE+K0
680 NEXT:C$=E$+MID$(C$,K(K-1))
690 RETURN
700 '** SUB: SPREAD BLANKS
710 '** SUB: SPREAD CHARACTERS
720 RETURN '** JUST IN CASE SPREAD IS CALLED
730 '** SUB FILL C$
740 IF L3>LEN(B$) THEN C$=B$:B$="":RETURN
750 FOR I=L3+1 TO 2 STEP-1
760  IF MID$(B$,I,1)=" "THEN IF MID$(B$,I-1,1)<>" " THEN 800
770  IF MID$(B$,I,1)="-"THEN IF I<=L3THEN C$=LEFT$(B$,I):B$=MID$(B$,I+1):RETURN
780 NEXT
790 PRINT"ERROR: LINE TOO LONG BETWEEN MARGINS. ";L3;"SPACES BETWEEN MARGINS. ":PRINT"LINE TO BE SPLI
800 C$=LEFT$(B$,I-1):B$=MID$(B$,I+1)
810 IF MID$(B$,1,1)=" " THEN B$=MID$(B$,2):GOTO810
820 C=LEN(C$):IF L3-C<6 OR SY=0 THEN RETURN
```

```
830 '** SYLLABICATION
840 D=INSTR(B$," "):IF D THEN D$=LEFT$(B$,D-1)ELSE D$=B$
850 IF RIGHT$(D$,1)="." THEN RETURN
860 D=LEN(D$):IF D<9 THEN RETURN
870 D1$=""
880 P1=0:FOR I=1 TO D:A=ASC(MID$(D$,I)):A=A+32*(95<A) '       ????
890    IF A<65 OR 95<A THEN 920
900    D1$=D1$+CHR$(A):P1=P1+A
910 NEXT
920 GOSUB 1080
930 D=LEN(D1$):
       IF D<9 THEN RETURN
940 GET 4,P1: E=INSTR(R$,D1$)
950 IF E=0
       THEN IF INSTR(R$,CHR$(126))>(238-D)
            THEN P1=P1+19:GOSUB1080:GOTO920
            ELSE 990
960 F1=E+D+1:E$=MID$(R$,F1,INSTR(F1,R$,CHR$(125))-F1)
970 FOR I=LEN(E$) TO 1 STEP -1:
       G=ASC(MID$(E$,I)):
       IF C+G<L3
          THEN C$=C$+" "+LEFT$(B$,G-1)+"-":B$=MID$(B$,G):RETURN
             ELSE NEXT
980 RETURN
990      PRINT:D2$="--> "+D1$+" <--":D2$=STRING$((48-LEN(D1$))/2," ")+D2$:
    PRINTD2$:PRINT"       IS NOT IN MY HYPHENATION DICTIONARY"
1000 PRINT"PLEASE TYPE IT IN WITH DASHES BETWEEN SYLLABLES."
1010 G$="":LINEINPUT"?";G$:IFG$=""THEN RETURN
1020 E$="":J=0:G=1
1030 H=INSTR(G,G$,"-"):
       IF H THEN E$=E$+CHR$(H-J):J=J+1:G=H+1:GOTO1030
1040 F$=D1$+CHR$(124)+E$+CHR$(125)+CHR$(126)
1050 G$=LEFT$(R$,INSTR(R$,CHR$(126))-1)
1060 IF LEN(G$)+LEN(F$)<244
       THENLSETR$=G$+F$:PUT4,P1:
       GET4,1:AB=AB+1:LSETAB$=MKI$(AB):PUT4,1:
       GOTO970
1070 P1=P1+19:GOSUB980:GOTO1060
1080 P1=(INT(P1)-51*FIX(INT(P1)/51))+1 : RETURN
1090 '******* SUB:MAKE CHARACTER SUBSTITUTIONS
1100 IF CH=0 THEN RETURN
1110 CI=0:
       FOR CG=1 TO CH
1120    CI=INSTR(CI+1,C$,MID$(CH$,CG,1))
1130    IF CI THEN MID$(C$,CI)=MID$(CI$,CG,1):GOTO1120
1140    NEXT:RETURN
```

```
1150 '********** SUB: HOF & HEADER
1160 IF L9=TM+1
        THEN 1180
        ELSE FOR L9=L9 TO P+TM:
        GOSUB 2080:
        NEXT:H1=H1+1
1170 IF WT THEN INPUT"WAIT -- ALIGN PAPER";ZZ$
1180 Z9=B1:GOSUB2680:L9=TM+1:IF H$="NO" THEN 1230
1190 IF H$=""THEN C$="" ELSE C$="- "
1200 C$=H$+C$+" PAGE"+STR$(H1)
1210 IF B1<1THENB1=1
1220 H4=L4:NJ=0:L4=0:GOSUB1980:L4=H4
1230 GG$=A$:X1=FL:X2=SY:X3=LM:LM=0:X4=T1:X5=T2:X6=KD:X7=KE:X8=KF
1240 OU=1:FORGG=0TO9:IFLEN(SH$(GG))THENA$=SH$(GG):GOSUB1720
1250 NEXT:OU=0
1260 A$=GG$:FL=X1:SY=X2:LM=X3:T1=X4:T2=X5:KD=X6:KE=X7:
        KF=X8:B1=Z9:GOSUB2690
1270 RETURN
1280 '******** SUB: COMMAND HANDLER  **********
1290 K7=0
1300 GOSUB1920:AC=ASC(MID$(A$,2)):IFAC>96THENMID$(A$,2,1)=
        CHR$(AC-32):MID$(A$,3,1)=CHR$(ASC(MID$(A$,3))-32)
1310 AC=ASC(MID$(A$,2)):IFAC>96THENMID$(A$,2,1)=CHR$(AC-32):
        MID$(A$,3,1)=CHR$(ASC(MID$(A$,3))-32)
1320 A9=INSTR(A9$,MID$(A$,1,3))/3
1330 ONA9GOTO1410,1440,1450,1470,1490,1510,1520,1530,1540,1560,
        1570,1580,1610,1620,1630,1640,1680,1690,1700,1720,1730,
        1740,1750,1770,1790,1800,1810,1820,1830,1860,1380,1350,
        1590,1600,1650,1660,1670,1420,1550
1340 PRINTMID$(A$,1,3)," IS A BAD COMMAND ON LINE ";Z$;:GOTO1890
.LS 1350 LM=0:L8=0
1360 L8=INSTR(L8+1,A$,","):IFL8 THEN LS(LM)=VAL(MID$(A$,L8+1)):
        LM=LM+1:GOTO1360
1370 LS(LM)=0:L8=0:GOTO1520
.RS 1380 RM=0:R8=0
1390 R8=INSTR(R8+1,A$,","):IFR8 THEN RS(RM)=VAL(MID$(A$,R8+1)):
        RM=RM+1:GOTO1390
1400 RS(RM)=0:R8=0:GOTO1530
.IT 1410 L4=VAL(MID$(A$,4,1)):L3=L3-L4:GOTO1890
.CMd 1420 IFMID$(A$,4,1)="?"THENPRINTMID$(A$,5)
1430 GOTO1890
.PL 1440 F=VAL(MID$(A$,4)):GOTO1890
.RP 1450 CLOSE F:Q=F:N1$=N2$(F)
1460 GOSUB2620:GOTO1890
.DFd 1470 N1$=MID$(A$,4):DF=1:
        IF (LEN(N1$))AND(N1$<>N2$(3))
         THENQ=3:GOSUB2620
1480 N2$(3)=N1$:GOTO1890
.MAm 1490 T3=VAL(MID$(A$,4)):
        IF T3
          THEN L1=L1-T3:L2=L2+T3
          ELSE L1=T1:L2=T2
1500 L3=L1-L2:GOTO1890
.WT1 1510 WT=VAL(MID$(A$,4)):GOTO1890
.LMm 1520 L2=VAL(MID$(A$,4)):T2=L2:L3=L1-L2:GOTO1890
.RMm 1530 L1=VAL(MID$(A$,4)):T1=L1:L3=L1-L2:GOTO1890
.HDd 1540 H$=MID$(A$,4):GOTO1900
.OV 1550 OV=VAL(MID$(A$,4)):GOTO1890
```

```
.CEd 1560 C$=STRING$(INT((L3-LEN(MID$(A$,4)))/2)," ")+MID$(A$,4):
          GOSUB2680:NJ=0:GOSUB1970:GOSUB2690:GOTO1900
.NF  1570 FL=1:GOTO1890
.FL  1580 FL=0:GOTO1890
.SMm 1590 I=VAL(MID$(A$,4)):MA$(I)=MID$(A$,5):GOTO1900
.EMm 1600 I=VAL(MID$(A$,4)):A$=LEFT$(A$,4)+MA$(I)+MID$(A$,5):GOTO1890
.[  n 1610 BM=VAL(MID$(A$,4)):GOTO1890
.TMm 1620 TM=VAL(MID$(A$,4)):GOTO1890
.JUM 1630 NJ=VAL(MID$(A$,4)):GOTO1890
.NJ  1640 NJ=0:GOTO1890
.NMm 1650 O3=L3:O4=L4:O2=L2:O1=L1:L2=VAL(MID$(A$,4)):
          L1=VAL(MID$(A$,INSTR(A$,",")+1)):L3=L1-L2:L4=0:
          GOTO1890
.SHd 1660 I=VAL(MID$(A$,4)):SH$(I)=". ST"+MID$(A$,5):GOTO1890
.OM  1670 L3=03:L4=04:L2=02:L1=01:GOTO1890  ?
.SPm 1680 B1=VAL(MID$(A$,4))-1:GOTO1890
.LFd 1690 C$=MID$(A$,4):K=1:K1=1:K4=L3-LEN(C$):GOSUB2680:NJ=0:
          GOSUB620:GOSUB1970:GOSUB2690:GOTO230
.TPm 1700 IF L9>P-BM-VAL(MID$(A$,4))
          THEN GOSUB 1160
     1710 GOTO1890
.ST$ 1720 F=F+1:N1$=MID$(A$,4):Q=F:GOSUB2620:GOTO1890
.PK  1730 POKEVAL(MID$(A$,4)),VAL(MID$(A$,INSTR(A$,",")+1)):GOTO1890
.WA  1740 PRINT"WAIT ";MID$(A$,4);:LINEINPUT" ";ZZ$:GOTO1890
.BLm 1750 B9=B1:B1=VAL(MID$(A$,4)):IF B1=0 THENB1=B9:GOTO1890
     1760 GOSUB2010:B1=B9:GOTO1890
.PAM 1770 P9=VAL(MID$(A$,4)):IF P9 THEN H1=P9-1
     1780 GOSUB1160:GOTO1890
.COd 1790 CO$(CO1)=MID$(A$,4)+CHR$(127)+STR$(H1):CO1=CO1+1:GOTO1890
.IN  1800 IN=1:OPEN"O",1,"INDEX":GOTO1890
.NI  1810 IN=0:GOTO1890
.N^< 1820 N1$=MID$(A$,4):Q=F:GOSUB2620:GOTO1900
.C.  1830 CG$=MID$(A$,4,1):CG=VAL(MID$(A$,5)):
          IF ASC(CG$)<>CG
          THEN CH$=CH$+CG$:CI$=CI$+CHR$(CG):CH=LEN(CH$):GOTO1890
     1840 CF=INSTR(CH$,CG$):IF CF=1
          THEN CH$=MID$(CH$,2):CI$=MID$(CI$,2)
          ELSE IF CF=CH AND CH<>0
               THEN CH$=LEFT$(CH$,CF-1):CI$=LEFT$(CI$,CF-1)
               ELSE CH$=LEFT$(CH$,CF-1)+MID$(CH$,CF+1):
                    CI$=LEFT$(CI$,CF-1)+MID$(CI$,CF+1$0,K_ZZN_"1@" @!MMGD!%]MGDMM!&C^^*
     1850 CH=LEN(CH$):GOTO1890
.SY1 1860 SY=VAL(MID$(A$,4)):CLOSE4:IF SY=0 THEN1890
     1870 OPEN"R",4,"DICTNARY":FIELD#4,245 AS R$,2 AS AB$,8 AS G$:
          GET 4,1
     1880 IF G$="DICTNARY"THEN AB=CVI(AB$)
          ELSELSETG$="DICTNARY":LSETR$=CHR$(125)+CHR$(126):
          LSETAB$=MKI$(0):
            FORM=1TO51:PUT4,M:NEXTM:GOTO1890
     1890 A9=INSTR(2,A$,"."):
          IF A9<>0
           THEN A$=MID$(A$,A9):GOTO1310
```

```
1900 B$="":GOTO230
1910 '**** SUB: RUN OUT B$
1920 IF B$=""THEN RETURN
1930 C$=B$:B$="":GOSUB2680:NJ=0
1940 IF(NF=0) AND (L9+B1<P)
        THEN GOSUB1980 ELSE GOSUB1970
1950 GOSUB2690:RETURN
1960 '***** SUB: PRINT A LINE *****
1970 IF L9>P-BM THEN GH$=B$:GK$=C$:GOSUB1160:B$=GH$:C$=GK$
1980 GOSUB1100
1990 GOSUB2140:IFLM=0 THEN L3=L3+L4
2000 L4=0:L9=L9+1:IFL9=0THENLPRINT""
2010 IF B1=0 THEN GOTO2060
2020 FOR B2=1 TO B1
2030  GOSUB 2080
2040  L9=L9+1
2050 NEXT
2060 C$="":RETURN
2070 '***** SUB: OUTPUT BLANK LINE TO CORRECT DEVICE
2080 ON OP GOSUB 2100,2110,2120
2090 GOTO2150
2100 LPRINT" ":RETURN
2110 PRINT#2,PP1;"'":PP1=PP1+10:RETURN
2120 PRINT:RETURN
2130 '**** SUB: SEND OUTPUT TO CORRECT DEVICE
2140 ON OP GOSUB 2170,2270,2280
2150 IF LM
        THEN IF LS(L8)=0 AND RS(L8)=0
                THEN LM=0
                ELSE L2=LS(L8):L1=RS(L8):L3=L1-L2:L8=L8+1
2160 RETURN
2170 LPRINTTAB(L2+L4+OV);:
       IF L4>0
       THEN L5=1
       ELSE L5=ABS(L4)+1
2180 LPRINT LEFT$(C$,L5);
2190 IF FL=0 THEN ON NJ GOTO 2200,2240,2210
2200 LPRINTMID$(C$,L5+1):RETURN
2210 'DELETED
2220 'DELETED
2230 'DELETED
2240 IF KF=0 THEN KF=1
2250 'DELETED
2260 DELETED
2270 PRINT#2,PP1;"'";STRING$(L2+L4+OV," ")+C$:PP1=PP1+1:RETURN
2280 PRINTTAB(L2+L4+OV);C$:RETURN
2290 '***** FINAL OUTPUT **********
2300 '*** CLOSE FILES
2310 IF IN THEN PRINT#1,"END"
2320 CLOSE#1
2330 '*** SORT AND PRINT INDEX
2340 IF IM=0 THEN GOTO2530
2350 IM=IM-1
```

```
2360 DIMIN$(IM):CLOSE#1:OPEN"I",1,"INDEX"
2370 FOR I=0TOIM:LINEINPUT#1,IN$(I):NEXT
2380 FORH=IM-1TO0STEP-1:N=H:
       FOR K=0 TO H:
        IF IN$(K)<IN$(N)
         THEN NEXT
         ELSE N=K: NEXT
2390 S$=IN$(H):IN$(H)=IN$(N):IN$(N)=S$:
     NEXT
2400 C$=STRING$(INT((L3-5)/2)," ")+"INDEX":GOSUB1970
2410 B1=0
2420 J=1
2430 IF J>IM THEN 2480
2440 J1=INSTR(IN$(J-1),CHR$(127)):J2=INSTR(IN$(J),CHR$(127))
2450 IF LEFT$(IN$(J-1),J1)<>LEFT$(IN$(J),J2)
       THEN 2470
       ELSE IF IN$(J-1)<>IN$(J)
           THEN IN$(J-1)=IN$(J-1)+MID$(IN$(J),J2)
2460 FORK1=JTOIM-1:
       IN$(K1)=IN$(K1+1):
     NEXT:IN$(IM)="":IM=IM-1
2470 J=J+1:GOTO2430
2480 FORJ=0TOIM-1
2490   C$=IN$(J)
2500   GOSUB1100:GOSUB1970:C$=""
2510 NEXT
2520 '***** PRINT TABLE OF CONTENTS
2530 IFCO1<1THENGOTO2600
2540 H$="":GOSUB1160:C$="TABLE OF CONTENTS":GOSUB1980
2550 FORJ=0TOCO1-1
2560 I=INSTR(CO$(J),CHR$(127))
2570 C$=LEFT$(CO$(J),I-1)+STRING$(L3-LEN(CO$(J)),46)+MID$(CO$(J),I+1)
2580 GOSUB1980
2590 NEXT
2600 GOSUB1160:RUN
2610 '***** SUB: OPEN FILES
2620 CLOSE#0:N2$(0)=N1$
2630 IF RIGHT$(N1$,1)=" "THENN1$=LEFT$(N1$,LEN(N1$)-1):GOTO2630
2640 IF LEFT$(RIGHT$(N1$,2),1)=":"THEN N1=VAL(RIGHT$(N1$,1))ELSE N1=0
2650 OPEN"I",0,N1$:LINEINPUT#0,G$
2660 IFG$<>"0 '" THENPRINTN1$;"  IS NOT AN ASCII FILE";:FORZ=1TO1000:NEXT:RUNELSE RETURN
2670 '***  SUB: BEEP AN ERROR
2680 U1=U1+1:NJ(U1)=NJ:RETURN
2690 NJ=NJ(U1):U1=U1-1:RETURN
```